PATENT
2080-3-66

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:
Yoon Seok Yang
Serial No:
Filed: Herewith
For: APPARATUS FOR ENCRYPTING/DECRYPTING REAL-TIME INPUT STREAM

Art Unit:

Examiner:

TRANSMITTAL OF PRIORITY DOCUMENT

Assistant Commissioner for Patents
Washington, D.C. 20231

Dear Sir:

Enclosed herewith is a certified copy of Korean patent application No. 2001-2644 which was filed on January 17, 2001 from which priority is claimed under 35 U.S.C. Section 119 and Rule 55.

Acknowledgment of the priority document(s) is respectfully requested to ensure that the subject information appears on the printed patent.

Respectfully submitted,

Date: January 16, 2002

By: _____

Jonathan Y. Kang
Registration No. 38,199
Amit Sheth
Registration No. P-50,176
Attorney for Applicant(s)

Lee & Hong
221 N. Figueroa Street, 11th Floor
Los Angeles, California 90012
Telephone: (213) 250-7780
Facsimile: (213) 250-8150

#2

대 한 민 국 특 허 청
KOREAN INTELLECTUAL PROPERTY OFFICE

별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto is a true copy from the records of the Korean Intellectual Property Office.

출 원 번 호 : 특허출원 2001년 제 2644 호
Application Number PATENT-2001-0002644

출 원 년 월 일 : 2001년 01월 17일
Date of Application JAN 17, 2001

CERTIFIED COPY OF PRIORITY DOCUMENT

출 원 인 : 엘지전자주식회사
Applicant(s) LG ELECTRONICS INC.

2001 년 11 월 14 일

특 허 청
COMMISSIONER

CERTIFIED COPY OF PRIORITY DOCUMENT

【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0003
【제출일자】	2001.01.17
【국제특허분류】	H04N
【발명의 명칭】	실시간 입력 스트림의 암호화/복호화 장치
【발명의 영문명칭】	Scrambler/descrambler of real time input stream
【출원인】	
【명칭】	엘지전자 주식회사
【출원인코드】	1-1998-000275-8
【대리인】	
【성명】	김용인
【대리인코드】	9-1998-000022-1
【포괄위임등록번호】	2000-005155-0
【대리인】	
【성명】	심창섭
【대리인코드】	9-1998-000279-9
【포괄위임등록번호】	2000-005154-2
【발명자】	
【성명의 국문표기】	양윤석
【성명의 영문표기】	YANG,Yoon Seok
【주민등록번호】	730408-1029512
【우편번호】	137-140
【주소】	서울특별시 서초구 우면동 2-13 청문빌라 404호
【국적】	KR
【심사청구】	청구
【취지】	특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사 를 청구합니다. 대리인 김용인 (인) 대리인 심창섭 (인)

【수수료】

【기본출원료】	19	면	29,000 원
【가산출원료】	0	면	0 원
【우선권주장료】	0	건	0 원
【심사청구료】	9	항	397,000 원
【합계】	426,000 원		

【첨부서류】 1. 요약서·명세서(도면)_1통

【요약서】

【요약】

실시간으로 입력되는 스트림을 AES 알고리즘을 이용하여 암호화하고 복호화하는 암호화/복호화 장치에 관한 것으로서, 특히 AES 알고리즘의 암호화 및 복호호화 과정을 하드웨어적으로 구현함으로써, 실시간으로 입력되는 스트림의 암호화와 복호화를 실시간으로 수행할 수 있다. 또한 상기 AES 알고리즘의 암호화와 복호화 과정을 하드웨어적으로 구현시 한 블록의 암호화나 복호화를 위한 키를 각 라운드마다 구하여 블록 라운드부로 출력함으로써, 블록 데이터의 암호화/복호화시 필요한 키 레지스터의 크기를 줄일 수 있으므로, 하드웨어의 크기를 줄임과 동시에 비용 절감의 효과가 있다.

【대표도】

도 2

【색인어】

AES, 암호화, 복호화

【명세서】

【발명의 명칭】

실시간 입력 스트림의 암호화 / 복호화 장치{Scrambler/descrambler of real time input stream}

【도면의 간단한 설명】

도 1은 본 발명에 따른 실시간 입력 스트림의 암호화/복호화 장치의 개념도

도 2는 본 발명에 따른 실시간 입력 스트림의 암호화/복호화 장치의 상세 블록도

도 3은 도 2의 키 스케쥴부의 상세 블록도

도 4는 도 2의 블록 라운드부의 암호화부의 상세 블록도

도 5는 도 2의 블록 라운드부의 복호화부의 상세 블록도

도 6은 도 2의 제어부의 상세 블록도

도면의 주요부분에 대한 부호의 설명

201 : 제어부 202 : 키 스케쥴부

203 : 블록 라운드부

【발명의 상세한 설명】

【발명의 목적】

【발명이 속하는 기술분야 및 그 분야의 종래기술】

<10> 본 발명은 실시간으로 입력되는 스트림을 AES(Advanced Encryption Standards) 알고리즘을 이용하여 암호화하고 복호화하는 암호화/복호화 장치에 관한 것이다.

<11> 최근 정보의 유료화와 개인 정보 보호의 중요성이 커짐에 따라 암호 및 복호기에 대한 중요성이 날로 커지고 있다. 특히 최근에는 그 동안 미연방 암호화 표준으로 사용되던 DES(Data Encryption Standard)를 대체할 차세대 암호화 표준이 새로이 제시되었다.

<12> NIST(미 상무부 기술 표준국)는 수년전부터 AES 후보 기술로 15개 알고리즘에 대한 평가 작업을 벌여 왔으며 지난해 중반 5개 기술로 범위를 좁힌 후 면밀한 검토 작업을 거친 끝에 차세대 암호 기술 표준으로 리즌댈(Rijndael) 알고리즘을 선정하고 세부 작업에 착수하였다. 상기 NIST는 일정 기간을 통해 상기 리즌댈 알고리즘을 공개, 검토하고 수정 작업을 거쳐 AES로 확정할 계획이다. 확정된 AES안은 연방 정보 처리 표준(FIPS)으로 선정된다.

<13> 상기 AES는 블록 암호화 알고리즘(즉, 스트림을 블록 단위로 모아 암호화를 수행함)으로서, 기존의 암호화 표준인 DES를 대체하며, 미국 정부 및 민간에 관한 정보를 보호하는데 쓰일 것이다. 그리고, 미 정부의 암호화 기기 수출의 허용과 맞물려 전 세계적으로 암호화 표준으로 널리 쓰일 전망이다. 1976년 미국 정

부에 의해 암호 표준으로 채택된 DES가 56비트 암호 체계를 사용하는 반면, AES는 128,192,256비트의 세가지 체계를 사용하여 암호화의 강도를 매우 높였으며, 성능, 효율성, 유연성, 구현 용이성이 뛰어난 장점을 가진다.

【발명이 이루고자 하는 기술적 과제】

<14> 그러나, 소프트웨어로 구현된 AES 알고리즘은 실시간으로 입력되는 스트림에 대해서는 암호화/복호화를 수행하지 못하는 단점이 있다. 즉, 실시간 동작을 위해, 블록 라운드는 다음의 블록 데이터가 전송되기 이전에 모든 라운드의 계산을 수행하여야 하는데, 소프트웨어로 이루어진 AES 알고리즘에서 블록 라운드는 이를 해결하지 못한다. 결국, 소프트웨어로 된 블록 라운드는 연속된 스트림이 입력될 때, 1 블록을 만들어 데이터를 처리하기까지의 시간을 확보하지 못하므로 연속되는 스트림에 대해서는 실시간 처리가 불가능하다.

<15> 본 발명의 목적은 AES 알고리즘으로 채택된 리즌델 알고리즘을 하드웨어로 구성함으로써, 실시간으로 입력되는 스트림을 실시간으로 암호화하고 복호화할 수 있도록 하는 암호화/복호화 장치를 제공함에 있다.

<16> 본 발명의 다른 목적은 리즌델 알고리즘의 하드웨어 구현시 공통적으로 사용되는 블록을 공유하여 단순한 하드웨어 구조로 암호화와 복호화를 모두 수행할 수 있는 암호화/복호화 장치를 제공함에 있다.

【발명의 구성 및 작용】

<17> 상기와 같은 목적을 달성하기 위한 본 발명에 따른 실시간 입력 스트림의 암호화/복호화 장치는, 바이트 단위의 실시간 스트림을 입력받아 블록 데이터로

변환한 후 암호화 또는 복호화를 위해 출력하고, 암호화 또는 복호화된 블록 데이터를 입력받아 바이트 단위로 변환하여 출력하는 제어부와, 외부에서 입력되는 블록의 크기와 키 값에 따라 키 스케쥴을 수행하여 각 라운드마다 암호화 또는 복호화를 위한 키 값을 출력하는 키 스케쥴부와, 상기 제어부로부터 블록 단위로 변환된 데이터와 상기 키 스케쥴부의 키 값을 입력받아 암호화 또는 복호화를 수행한 후 상기 제어부로 출력하는 블록 라운드부를 포함하여 구성되는 것을 특징으로 한다.

<18> 상기 블록 라운드부는 상기 제어부로부터 다음 블록 데이터가 입력되기 전까지 현재 암호화 또는 복호화되는 데이터의 모든 라운드 계산을 완료한 후 상기 제어부로 출력하는 것을 특징으로 한다.

<19> 상기 키 스케쥴부는 상기 블록 라운드부에서 각 라운드 처리를 위해 필요한 키를 각 라운드마다 키 스케쥴하여 상기 블록 라운드부로 출력하는 것을 특징으로 한다.

<20> 상기 키 스케쥴부는 실제 한 라운드에서 필요한 키 값만큼의 키 레지스터로 구성되는 것을 특징으로 한다.

<21> 상기 제어부는 각 라운드의 키 값을 각 라운드마다 생성하기 위해 제어 신호를 생성한 후 상기 키 스케쥴부로 출력하는 것을 특징으로 한다.

<22> 본 발명의 다른 목적, 특징 및 잇점들은 첨부한 도면을 참조한 실시예들의 상세한 설명을 통해 명백해질 것이다.

<23> 이하, 본 발명의 바람직한 실시예를 첨부도면을 참조하여 상세히 설명한다.

<24> 도 1은 본 발명에 따른 AES 암호화/복호화 장치의 개략도로서, 입출력 신호들의 예를 보이고 있고, 도 2는 도 1의 암호화/복호화 장치의 상세 블록도이다.

<25> 도 2를 보면, 본 발명에 따른 AES 암호화/복호화 장치는 크게 128/192/256 비트 키를 입력받아 키 스케쥴을 담당하는 키 스케쥴부(202), 128비트 블록 데이터를 입력받아 암호화 또는 복호화를 수행하는 블록 라운드부(203), 및 상기 키 스케쥴부(202), 블록 라운드부(203)에 필요한 제어 신호를 생성하고, 바이트 단위의 스트림을 입력받아 입력 버퍼를 통해 블록 단위로 변환한 후 상기 블록 라운드부(203)로 출력하거나, 상기 블록 라운드부(203)에서 출력되는 블록 데이터를 바이트 단위로 변환하여 외부로 출력하는 제어부(201)로 구성된다.

<26> 이때, 상기 제어부(201)로 입력되는 신호에는 바이트 단위의 암호화 또는, 복호화를 위한 스트림 데이터(MPEG 시스템 스트림, DSS 스트림, 기타 스트림) Data_in[7:0], 데이터 밸리드 신호 Data_valid, 키 값의 크기를 알려주는 wsel[1:0] 신호, 상기 블록 라운드부(203)에서 암호화 또는 복호화된 블록 데이터 Out_block[127:0], 암호화 또는 복호화된 블록 데이터 밸리드 신호 Out_block_valid가 있다. 그리고, 상기 제어부(201)에서 출력되는 신호에는 키 스케쥴부(202)와 블록 라운드부(203)를 거쳐 암호화 또는, 복호화된 바이트 단위 스트림 데이터 Out_data[7:0]와 출력 데이터 밸리드 신호 Out_valid, 암호화 또는 복호화를 위해 상기 블록 라운드부(203)로 출력되는 블록 데이터 In_block[127:0], 암호화 또는 복호화를 위한 블록 데이터 밸리드 신호 In_block_valid, 및 상기 키 스케쥴부(202)로 출력되는 키 스케쥴 시작 제어 신호 Key_start가 있다.

<27> 또한, 상기 키 스케쥴부(202)로 입력되는 신호에는 키 값의 크기를 알려주는 wsel[1:0], 암호(encrypt)인지 해독(decrypt)인지를 알려주는 Encrypt_en 신호, Key_data[128,192,256], 그리고 상기 제어부(201)에서 출력되는 Key_start 신호가 있으며, 상기 키 스케쥴부(202)에서 출력되는 신호는 암호화 또는 복호화를 위한 키 데이터 Round_key/I_round_key[127:0], 키 데이터 밸리드 신호 Round_key_valid가 있다.

<28> 상기 블록 라운드부(203)로 입력되는 신호에는 키 값의 크기를 알려주는 wsel[1:0], 암호(encrypt)인지 해독(decrypt)인지를 알려주는 Encrypt_en 신호, 상기 제어부(201)에서 출력되는 암호화 또는 복호화를 위한 블록 데이터 In_block[127:0], 상기 블록 데이터 밸리드 신호 In_block_valid, 상기 키 스케쥴부(202)에서 출력되는 암호화 또는 복호화를 위한 키 데이터 Round_key/I_round_key[127:0], 키 데이터 밸리드 신호 Round_key_valid가 있으며, 상기 블록 라운드부(203)에서 출력되는 신호에는 암호화 또는 복호화된 블록 데이터 Out_block[127:0], 상기 블록 데이터 밸리드 신호 Out_block_valid가 있다.

<29> 여기서, 상기 평문 즉, 암호화를 위해 입력되는 블록 데이터는 128로 고정되어 있다. 즉, 상기 평문은 암호화 대상이 되는 블록 데이터이다. 상기 평문과 키 값에 따른 wsel 신호 값을 하기의 표 1에 나타내었다.

<30>

【표 1】

wsel[1:0]	평문=128
키 값=128	00
키 값=192	01
키 값=256	10

<31> 일 예로, 입력되는 wsel[1:0] 신호가 01이라고 가정하면, 암호화될 평문 즉, 블록의 크기는 128이고, 이때의 키 값은 192임을 의미하며, 이러한 wsel[1:0] 신호가 상기 제어부(201), 키 스케쥴부(202), 및 블록 라운드부(203)로 동시에 입력된다.

<32> 도 3은 상기 키 스케쥴부(202)의 구성 블록도로서, 키 확장부(301)와 키 선택부(302)로 구성된다.

<33> 먼저, 상기된 표 1과 같이 wsel 신호에 의해 키 값의 크기가 결정되어 키 확장부(301)로 입력된다. 그러면, 상기 키 확장부(301)는 키 확장 과정(key_expans)을 통해 상기 입력된 키 값을 블록의 크기 * (라운드의 수 + 1)만큼의 크기로 확장한다. 그리고, 각 확장된 키는 키 선택부(302)를 통해 각 라운드마다 블록 라운드부(203)로 입력되어 암호화/복호화 즉, 사이퍼/디사이퍼(cipher/decipher) 리즌델 알고리즘이 수행된다. 상기 키 선택부(302)는 각 라운드마다 필요한 128비트 키를 선택하여 상기 블록 라운드부(203)로 출력한다. 이는 블록 라운드부(203)에서 암호화 또는 복호화를 수행할 때 각 라운드마다 키 값이 필요하기 때문이다.

<34> 즉, 상기 키 스케쥴부(202)는 암호화나 복호화를 위한 키를 각 라운드마다 구하여 블록 라운드부(203)로 출력한다. 만일, 현재 암호화 과정이라면 키 선택

부(302)는 각 라운드마다 필요한 키 값 Round_key[127:0]을 선택적으로 생성하여 출력하고, 복호화 과정이라면 초기 과정에서 미리 암호화 키 스케쥴을 통해 구해진 최종 128비트 키 값을 입력받아 암호화의 역 동작을 수행하여 각 라운드마다 필요한 키 값을 생성한다.

<35> 이때, 한 블록의 암호화나 복호화를 위한 키를 각 라운드마다 구하지 않고 확장된 키를 미리 저장한다면 블록의 크기 * (라운드의 수 + 1)만큼의 키 레지스터가 필요하게 된다. 일 예로, 블록의 크기가 128비트이고, 키 값의 크기가 256비트이면 라운드의 수는 14가 되므로 필요한 키 레지스터의 크기는 128*(14+1)비트 = 1920비트 = 176바이트이다. 또 다른 예를 들면, 블록의 크기가 128비트이고, 키 값의 크기가 128비트이면 라운드의 수=10이 되므로 필요한 키 레지스터의 크기는 128*(10+1)비트 = 1408비트 = 176바이트이다.

<36> 즉, 키 스케쥴에 의해 확장된 키 값(블록 크기 * 라운드의 수)을 모두 가지도록 키 레지스터를 구성한다면 매우 많은 양의 키 레지스터가 필요하게 된다. 이는 키 레지스터의 불필요한 낭비를 초래한다.

<37> 따라서, 본 발명에서는 이러한 키 레지스터의 낭비를 막기 위하여 한 블록의 암호화나 복호화를 위한 키를 각 라운드마다 구하여 블록 라운드부(203)로 출력한다. 이렇게 함으로써, 실제 한 라운드에서 필요한 키 값만큼의 레지스터만 필요하게 된다.

<38> 일 예로, 블록의 크기가 128비트이고, 키 값의 크기가 256비트일 때 필요한 키 레지스터는 블록의 크기 * 한 라운드의 값만 있으면 되므로, 필요한 키 레지스터의 크기는 128*1 = 16바이트이다.

<39> 이와 같이, 본 발명의 암호화/복호화 장치는 키 레지스터의 크기를 줄이기 위하여 각 라운드마다 키 스케쥴을 하여 암호화/복호화 동작을 수행한다. 그리고, encrypt_en 신호와 하나의 키 스케쥴부(202)를 이용하여 암호화와 복호화를 모두 수행할 수 있는 장점을 지닌다.

<40> 한편, 도 4와 도 5는 상기 블록 라운드부(203)의 상세 블록도로서, 도 4는 암호화시 이용되는 암호화부(400)의 구성 블록도이고, 도 5는 복호화시 이용되는 복호화부(500)의 구성 블록도이다.

<41> 즉, 상기 블록 라운드부(203)는 상기 제어부(201)의 입력 버퍼로부터 제공된 블록 데이터 In_block[127:0]와 밸리드 신호 In_block_valid를 이용하여 암호화/복호화 동작을 수행한다.

<42> 이때 각 라운드마다 필요한 암호화키/복호화키는 상기 키 스케쥴부(202)에서 입력받는다. 또한, 상기 블록 라운드부(203)는 제어부(201)로부터 다음의 블록 데이터가 전송되기 이전에 모든 라운드의 계산을 수행한다. 이는 상기 제어부(201), 키 스케쥴부(202), 및 블록 라운드부(203)가 게이트로 구성한 하드웨어 구조이므로 실시간 처리가 가능하기 때문이다.

<43> 만일, 현재 암호화를 수행한다면, 상기 도 4의 암호화부(400)의 데이터 변환부(401)는 상기 제어부(201)로부터 암호화할 블록 데이터 In_block[127:0]를 입력받아 바이트 단위로 변환한다. 여기서, 상기 데이터 변환부(401)는 비선형적 특성을 가지므로, 바이트 단위로 변환할 때 비선형적 치환을 수행하며, 암호화 알고리즘에서 암호화의 강인성(robust)을 결정하는 중요한 역할을 한다.

<44> 상기 데이터 변환부(401)에서 바이트 단위로 치환된 데이터는 쉬프터(402)로 입력되어 로우 방향으로 쉬프트된 다음 믹서(403)로 입력되어 컬럼 방향으로 믹스된다. 그리고 나서, 키 혼합기(404)로 입력되어 상기 키 스케쥴부(202)에서 출력되는 라운드 키 Round_key[127:0]와 합쳐진 후 제어부(201)로 출력된다. 즉, 상기 입력되는 블록 데이터가 데이터 변환부(401), 쉬프터(402), 믹서(403), 및 키 혼합기(404)를 순차적으로 거치면 한 라운드가 되며, 기 설정된 라운드 수에 따라 상기 과정을 반복한다. 이때 반복되는 라운드의 수는 키 값에 의해 결정된다.

<45> 만일, 현재 복호화를 수행한다면 상기된 도 4의 암호화의 역 과정으로 수행하면 되고, 이때의 하드웨어 구성은 도 5와 같다.

<46> 즉, 상기 제어부(201)에서 복호화를 위해 블록 데이터 In_block[127:0]가 입력되면 키 혼합기(501)는 상기 키 스케쥴부(202)에서 출력되는 인버스 라운드 키 I_round_key와 더하여 인버스 믹서(502)로 출력한다. 상기 인버스 믹서(502)는 상기 키 혼합기(501)에서 출력되는 데이터의 컬럼 방향으로 인버스 믹스한 후 인버스 쉬프터(503)로 출력하여 로우 방향으로 인버스 쉬프트를 수행한다. 이는 암호화시 컬럼 방향으로 믹스를 하고, 로우 방향으로 쉬프트를 하였기 때문이며, 상기 암호화시와 반대로 하면 된다. 상기 인버스 쉬프터(503)에서 출력되는 데이터는 데이터 변환부(504)를 거쳐 상기 제어부(201)로 출력된다.

<47> 이와 같이, 상기 블록 라운드부(203)의 암호화부(400)에서 암호화 또는 복호화부(500)에서 복호화된 블록 데이터 Out_block[127:0]는 블록 데이터 출력 밸리드 신호 Out_block_valid와 함께 제어부(201)에 있는 출력 버퍼로 전달된다.

<48> 도 6은 상기 제어부(201)의 상세 블록도로서, 암호화 또는 복호화될 비트 스트림을 입력받아 블록 단위로 변환한 후 변환된 블록 데이터 In_block[127:0]와 블록 데이터 밸리드 신호 In_block_valid를 상기 블록 라운드부(203)로 출력하는 입력 버퍼(601), 상기 블록 라운드부(203)에서 암호화 또는 복호화된 데이터 Out_block[127:0]와 데이터 밸리드 신호 Out_block_valid를 입력받아 바이트 단위로 출력하는 출력 버퍼(603), 및 상기 입/출력 버퍼(601,603)의 데이터 저장 및 출력을 제어하며, 각 라운드의 키 값을 각 라운드마다 생성하기 위해서 제어 신호 key_start를 생성하여 키 스케쥴부(202)로 출력하는 상태 제어부(602)로 구성된다.

<49> 즉, 상기 제어부(201)의 입력 버퍼(601)는 암호화 또는 복호화를 위해 바이트 단위로 입력되는 비트스트림 Data_in[7:0]을 저장한 후, 상기 저장된 비트 스트림 Data_in[7:0]이 한 블록 크기만큼 모아지면 이를 블록 단위로 상기 블록 라운드부(203)에 출력한다. 상기 상태 제어부(602)는 입/출력 버퍼(601,603)의 데이터 저장 및 출력을 제어하며, 또한 각 라운드의 키 값을 각 라운드마다 생성하기 위해서 키 스케쥴부(202)에 제어 신호 key_start를 생성하여 출력한다. 그리고, 상기 출력 버퍼(603)는 상기 블록 라운드부(203)에서 암호화나 복호화된 데이터를 입력받아 저장하여 다시 바이트 단위의 데이터 Out_data[7:0]를 밸리드 신호 Out_valid와 함께 출력한다.

【발명의 효과】

<50> 이상에서와 같이 본 발명에 따른 실시간 입력 스트림의 암호화/복호화 장치에 의하면, AES 알고리즘의 암호화 및 복호화 과정을 하드웨어적으로 구현함으로

써, 실시간으로 입력되는 스트림의 암호화와 복호화를 실시간으로 수행할 수 있다. 또한 본 발명은 실시간 암호화가 필요한 모든 정보 암호화 기기에 이용될 수 있으며 특히 제한 수신 시스템을 위한 유료화 스트림 암호화 및 개인 정보 암호화 등에 폭넓게 이용되는 효과를 볼 수 있다.

<51> 그리고, 상기 AES 알고리즘의 암호화와 복호화 과정을 하드웨어적으로 구현시 한 블록의 암호화나 복호화를 위한 키를 각 라운드마다 구하여 블록 라운드부로 출력함으로써, 블록 데이터의 암호화/복호화시 필요한 키 레지스터의 크기를 기존보다 최대 1/15 ~ 최소 1/10로 줄여 하드웨어의 크기를 줄임과 동시에 비용 절감의 효과가 있다.

<52> 이상 설명한 내용을 통해 당업자라면 본 발명의 기술 사상을 일탈하지 아니하는 범위에서 다양한 변경 및 수정이 가능함을 알 수 있을 것이다.

<53> 따라서, 본 발명의 기술적 범위는 실시예에 기재된 내용으로 한정되는 것이 아니라 특허 청구의 범위에 의하여 정해져야 한다.

【특허청구범위】

【청구항 1】

바이트 단위의 실시간 스트림을 입력받아 블록 데이터로 변환한 후 암호화 또는 복호화를 위해 출력하고, 암호화 또는 복호화된 블록 데이터를 입력받아 바이트 단위로 변환하여 출력하는 제어부;

외부에서 입력되는 블록의 크기와 키 값에 따라 키 스케쥴을 수행하여 각 라운드마다 암호화 또는 복호화를 위한 키 값을 출력하는 키 스케쥴부; 그리고

상기 제어부로부터 블록 단위로 변환된 데이터와 상기 키 스케쥴부의 키 값을 입력받아 암호화 또는 복호화를 수행한 후 상기 제어부로 출력하는 블록 라운드부를 포함하여 구성되는 것을 특징으로 하는 실시간 입력 스트림의 암호화/복호화 장치.

【청구항 2】

제 1 항에 있어서, 상기 제어부는

외부로부터 입력되는 바이트 단위의 실시간 스트림을 저장한 후 상기 외부에서 입력되는 크기의 블록 데이터로 변환하여 상기 블록 라운드부로 출력하는 입력 버퍼와,

상기 블록 라운드부에서 암호화 또는 복호화된 블록 데이터를 입력받아 바이트 단위로 변환한 후 외부로 출력하는 출력 버퍼를 포함하여 구성되는 것을 특징으로 하는 실시간 입력 스트림의 암호화/복호화 장치.

【청구항 3】

제 2 항에 있어서, 상기 블록 라운드부는

상기 제어부로부터 다음 블록 데이터가 입력되기 전까지 현재 암호화 또는 복호화되는 데이터의 모든 라운드 계산을 완료한 후 상기 제어부의 출력 버퍼에 저장하는 것을 특징으로 하는 실시간 입력 스트림의 암호화/복호화 장치.

【청구항 4】

제 1 항에 있어서, 상기 키 스케쥴부는

상기 블록 라운드부에서 각 라운드 처리를 위해 필요한 키를 각 라운드마다 키 스케쥴하여 상기 블록 라운드부로 출력하는 것을 특징으로 하는 실시간 입력 스트림의 암호화/복호화 장치.

【청구항 5】

제 4 항에 있어서, 상기 키 스케쥴부는

입력되는 키 값을 (블록의 크기 * (라운드의 수 + 1))만큼의 크기로 확장하는 키 확장부와,

상기 확장된 키 값으로부터 각 라운드마다 필요한 128비트 키를 선택하여 상기 블록 라운드부로 출력하는 키 선택부를 포함하여 구성되는 것을 특징으로 하는 실시간 입력 스트림의 암호화/복호화 장치.

【청구항 6】

제 1 항에 있어서, 상기 키 스케쥴부는

입력되는 키 값을 (블록의 크기 * (라운드의 수 +1))의 크기로 확장한 후 각 라운드마다 필요한 128비트의 키 값을 선택하는 과정을 하나의 키 레지스터를 이용하여 수행하는 것을 특징으로 하는 실시간 입력 스트림의 암호화/복호화 장치.

【청구항 7】

제 6 항에 있어서, 상기 키 스케쥴부는

실제 한 라운드에서 필요한 키 값만큼의 키 레지스터로 구성되는 것을 특징으로 하는 실시간 입력 스트림의 암호화/복호화 장치.

【청구항 8】

제 7 항에 있어서, 상기 키 레지스터는

입력되는 블록의 크기 * 한 라운드의 크기의 용량을 갖는 것을 특징으로 하는 실시간 입력 스트림의 암호화/복호화 장치.

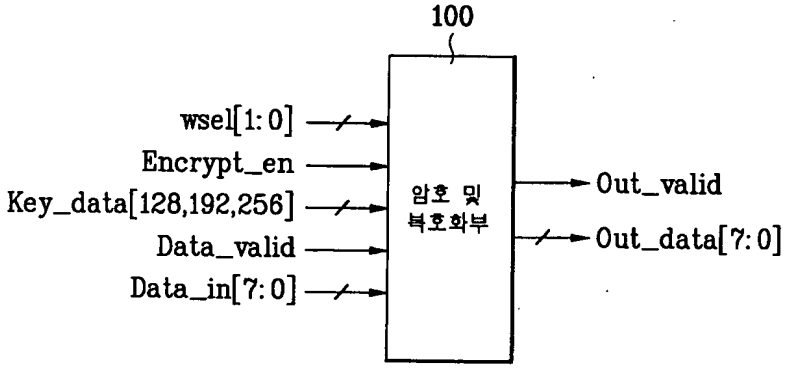
【청구항 9】

제 1 항에 있어서,

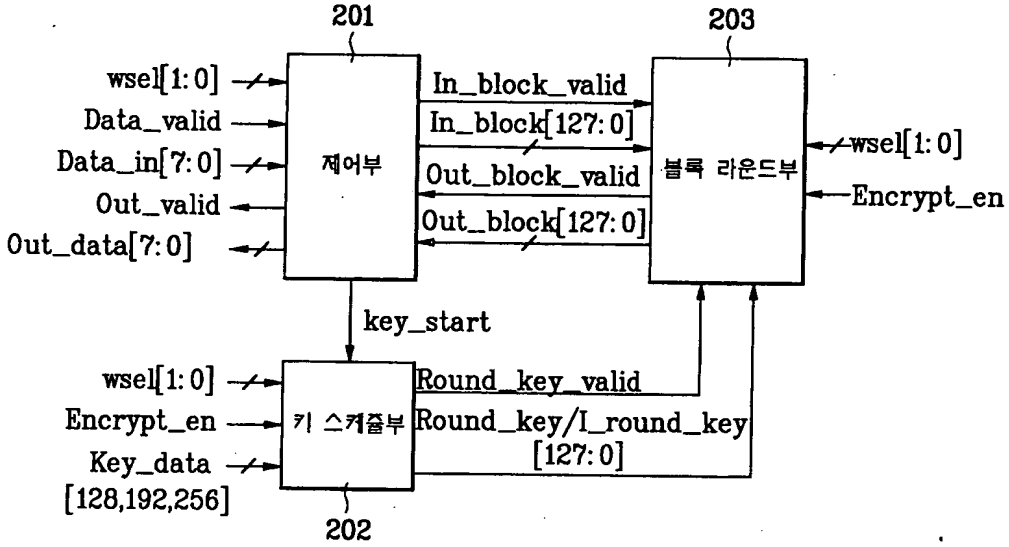
상기 제어부는 각 라운드의 키 값을 각 라운드마다 생성하기 위해 제어 신호를 생성한 후 상기 키 스케쥴부로 출력하는 것을 특징으로 하는 실시간 입력 스트림의 암호화/복호화 장치.

【도면】

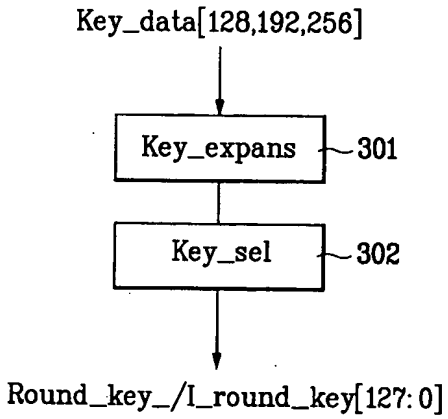
【도 1】


100
wsel[1:0]
Encrypt_en
Key_data[128,192,256]
Data_valid
Data_in[7:0]
암호 및 복호화부
Out_valid
Out_data[7:0]

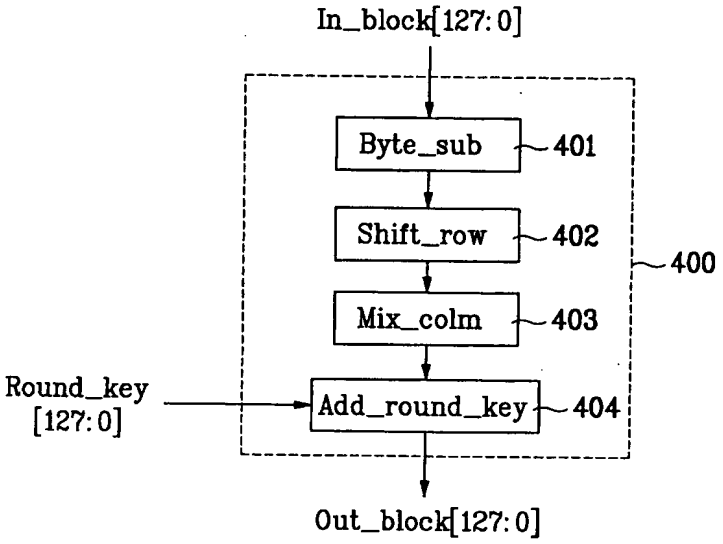
【도 2】


201
203
wsel[1:0]
Data_valid
Data_in[7:0]
Out_valid
Out_data[7:0]
제어부
In_block_valid
In_block[127:0]
Out_block_valid
Out_block[127:0]
블록 라운드부
wsel[1:0]
Encrypt_en
key_start
wsel[1:0]
Encrypt_en
Key_data
[128,192,256]
키 스케줄부
Round_key_valid
Round_key/I_round_key
[127:0]
202

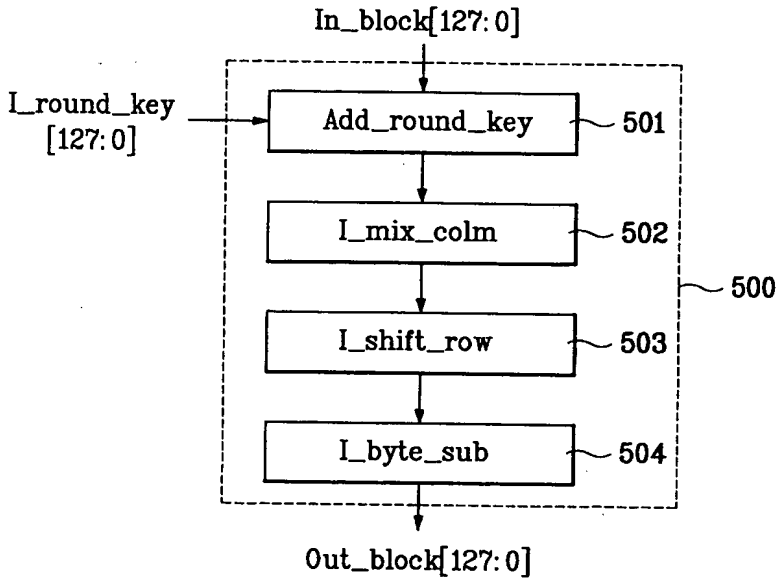
【도 3】


Key_data[128,192,256]
Key_expans
301
Key_sel
302
Round_key_/I_round_key[127:0]

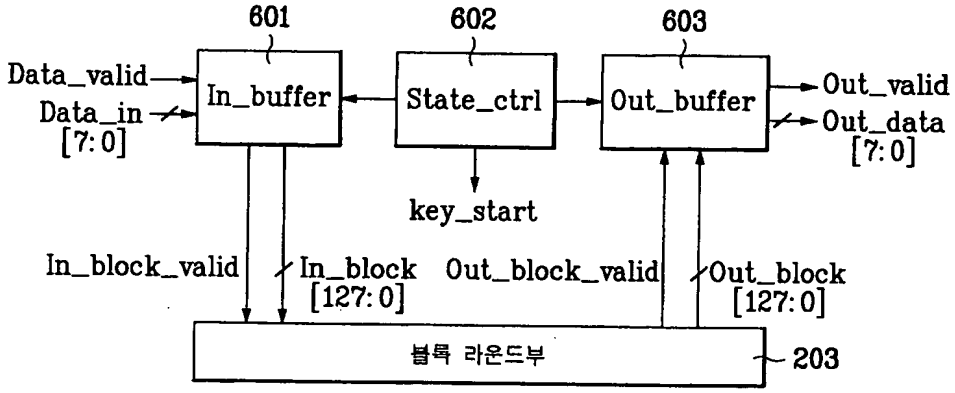
【도 4】


In_block[127:0]
Byte_sub
401
Shift_row
402
Mix_colm
403
400
Round_key
[127:0]
Add_round_key
404
Out_block[127:0]

【도 5】


In_block[127:0]
I_round_key
[127:0]
Add_round_key
501
I_mix_colm
502
500
I_shift_row
503
I_byte_sub
504
Out_block[127:0]

【도 6】


601
602
603
Data_valid
Data_in
[7:0]
In_buffer
State_ctrl
Out_buffer
Out_valid
Out_data
[7:0]
key_start
In_block_valid
In_block
[127:0]
Out_block_valid
Out_block
[127:0]
블록 라운드부
203